MOTD
The International Newsletter of the OS-9 Users Group
U S $3.50 Canada $4.25
Mar/April 1989

CONTENTS:

Sun Mar 26, 1989
D.KALEITA at 17:49 EST

I finally have the results of the OS-9 Users Group Elections. Only 46 members sent in their ballot (out of almost 800 members), but that's enough to pick new officers. The results are:

PRESIDENT:
Kevin Darling 36 votes / Bert Schneider 10 votes

VICE PRESIDENT:
Bruce Isted - 30 votes / Greg Law - 16 votes

SECRETARY:
Mark Griffith - 18 votes / Bill Turner - 28 votes

TREASURER:
George Dorner - 45 votes / <abstain> - 1 vote

The next step is for the new President to announce who will be Editor, Librarian and Director-at-large during his tenure. I'll let everyone know who Kevin chooses as soon as he makes it official.

It's been an experience. <Dave Kaleita>

Dave has served the UG long and well.. Most will join me in saying: Thanks Dave!

From the Editor

As usual, many many things are happening. The biggest news is the announcement of RAVE, and its release on May 15th. Although RAVE may never see implementation on the Personal System level, its advent provides a standard for Applications programmers to work to. Also, for the first time, OS-9 has a graphics extension. Although details are not contained in the MOTD overview, the basic capabilities are outlined: clipboard, resource, and font management to name a few. RAVE is just one of several good news type announcements that I expect this year... stay tuned!

The other biggie is the election results. The vote count may seem small, but it's a lot bigger than last time, and may be the biggest turnout yet.

OS-9 is alive and well, and so is our UG, despite the lack of a column from any of the current officers. I guess everyone is just overwhelmed right now.

In order to reduce publishing costs, I have gone

Continued Page 8

SPECIAL Microware EDITION

Well..... mostly

RAVE
Real-Time Audio/Video Environment

1.0 Product Overview

RAVE (Real-Time Audio/Video Environment) is a multimedia development tool and user interface that greatly simplifies the design of realistic man/machine interfaces for real-time process control systems. As an extension of the OS-9 real-time operating system, RAVE enables designers to combine high-quality audio and video, computer generated graphics and customizable menus in the same user interface. With RAVE, designers can quickly configure realistic user interfaces and control panels using real-world sounds and images. Because the resulting user interface better represents the actual control environment, it can be manipulated and understood by non-technical users.

2.0 The Evolution of Man/Machine Graphic Interfaces

Historically, the user interfaces available for factory floor process control applications have been difficult to use, and in some cases, intimidating. These interfaces typically consisted of non-interactive displays or light boards. At best, they were based on black and white graphics drawn on a conventional CRT. In the early 1980's user interfaces such as GKS and VDI were developed and popularized. Users could now hope to find at least color drawings representing actual devices that the system controlled. Audio capabilities, if any, were pretty much limited to "beeps" and "buzzes" and other non-descriptive noises. To compound the problem, these emerging interfaces were designed for UNIX or IBM Personal Computer systems without any consideration for real-time software requirements. Because these standards had not been integrated with a real-time operating system or kernel, factory floor systems were still largely limited to 1960's- style user interfaces

Furthermore, these interfaces were still hard to use and understand by non- technical users. Simply put, these standards were designed by programmers for use by programmers. And because programmers have not always had the time or resources to create artistic masterpieces, graphic images left much to be desired in the realm of visual understanding and recognition. All of this resulted in factory control systems that could not be easily understood, let alone actively manipulated by non-technical users.

3.0 The RAVE User Interface

Recognizing this complete lack of an easy-to-use and easy-to-understand man/machine interface for real-time process control systems, Microware has developed RAVE (Real-Time Audio/Video Environment). RAVE was developed on the simple premise that non-technical users can intuitively understand real-world visual representation--or pictures---far easier than basic drawn image. Now, with RAVE the tools for user interface development are a microphone and camera, rather than a mouse and keyboard. Not only does RAVE support high-resolution, real-world pictures, but it also offers full-motion video and CD-quality sound. These advanced features allow a process or model to be represented by its natural image and natural sounds, as opposed to a programmer's personal conception of how that item should be represented.

RAVE consists of three packages:

- The Graphic File Manager
- Graphics Support Library
- The Presentation Editor

The Graphics File Man-

Continued Page 4

RAINBOWFEST CHICAGO 1989 by Kevin Darling

This is the fest as I saw it... being in a booth most of the time, my viewpoint is probably much different than usual. I hope other people who attended will also upload what they saw and thought of things. I will just ramble through:

Marsha and I got in Thursday afternoon after driving 14 hours from North Carolina. Before bed, we went across the street to one of the biggest malls in the country. I mention this mall only because we and several others dropped into a new hologram store there... and one of the dozens of pictures was the most amazing we'd ever seen. It was laying horizontally, and as you walked up to it, a full-sized lab microscope sprang into 3-D view, sticking out about a foot. You could even look into the greenish ghost-like lens and see a slide through it! Incredible. I think this hologram was $200, btw <grin>.

Friday around noon, others started to show up: Kent Meyers, Tony DiStefano and Chris Rochon, Dale Puckett (without Esther!), and Gary Robinson from Tandy Towers with his wife Karen (at their first fest).

The show was open from 7pm to 10pm Friday night. Gary, Karen, Marsha and I wandered through in traditional first-night fashion... you grab a copy of all the literature to look through later in your room when you have time <grin>. Altho Lonnie said that attendance was slightly down from last year, I couldn't tell. There seemed to be more booths to me, and thousands of people around as always. At fests, most everyone keeps their eyes glancing at everyone else's name badges, hoping to spot someone you know only by name on a forum.You meet a lot of people that way. (Kent wears a jacket with his name on the back. He meets even more friends with this trick!) The Chicago fest is always easier to get around in than Princeton, as the room is a little larger.

Above the noise of people talking, the major background sound is the music of MIDI synths playing from several booths. Plus the Rainbow public address, which everyone hates because it totally prevents conversation when they blare out something about "Get your picture taken with CoCo Cat!", or the like. Every year we all swear we're gonna sabotage the PA system....

Many of the CoCos on display had been built into PC-clone cases. It's almost becoming hard to find a plain coco at a fest. Most of the cased ones were running OS-9, while most of the normal ones were running RSDOS adventure games. Someday a smart person is going to bring out a simple-to-do CoCo case kit and make millions at one of these fests, I hope.

Friday nights go by quickly, and a bunch of us ended up in Kent's room (where I keep my computer). Those who haven't seen tons of VEF pictures get to see them there, plus people bring in programs that they've been working on to get opinions and suggestions. We also hash out ideas every year in Kent's room for new hardware, etc. Mike Haaland had sent along a gfx editor he was writing for us to look at this time, and it was pretty impressive with its use of pulldown tool menus.

-o-

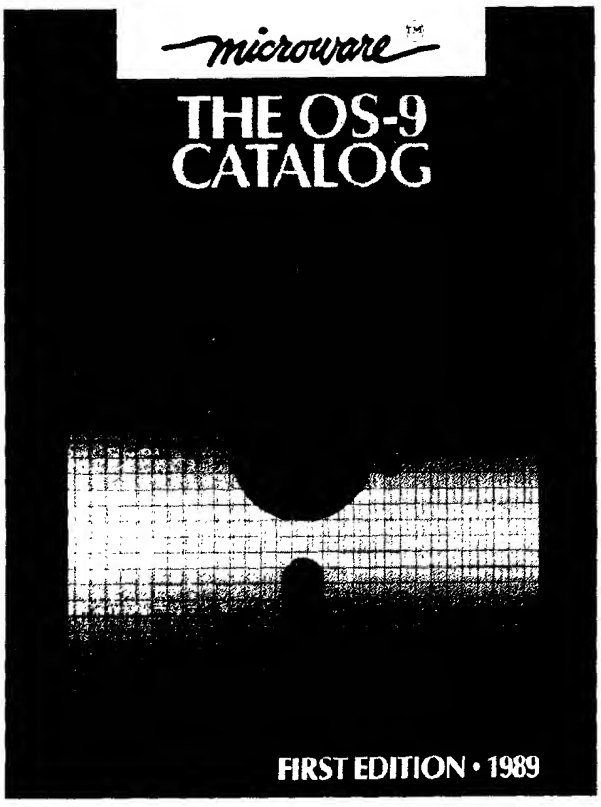
Saturday morning we finally had permission to use Frank Hogg's booth for the OS-9 Users Group (Frank couldn't come because of a back injury, but he decided to donate the booth to the UG - thanks, Frank!), and we opened it up about 11am. Over the weekend quite a few forum people stopped by to say hi; and we also gained about 30 new members. We had copies of Dibble's new OS-9/68000 Insights book and the new MW OS-9 Catalog to look at. **The Catalog and also the MW Sourcebook are free** for the asking just by calling MW in Des Moines. Dibble's book is about $40, also from Microware. All are recommended reading material.

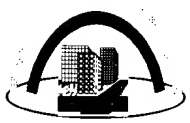
Paul Ward and his friend Mark Sheffield, along with Marsha and Dick White,helped man the booth. Paul donated a lot of his "Start OS-9" books to the booth, with proceeds going to the UG. Thanks, Paul!

Saturday at noon was when my "seminar" was. I had originally intended to demo some of the forum programs there. But I noticed that they had assigned me one of the bigger rooms, and I remembered from last year that no one could see a monitor that far away (they REALLY need to give us some big screen monitors or several tied thru an amplifier!). Plus most seminars put people to sleep when things start getting over their head. Thus I decided to try something radically different at the last moment (those there will attest to that :-).So I opened it up as a bull session, and the topics ranged all over the spectrum. Some of it was really enlightening to us. People apparently had a lot of opinions, and have wanted to speak out for a long time.

Questions and answers covered: What a coco-4 should be (dual processor6809/68000); Would they move to OSK? (not unless they could run their OS-9 software with another cpu or emulator); What software do people want? (cheaper and easier word processors, spreadsheets, business software, educational software, more business software, all under OS-9); Is everyone addicted to the CLEAR key? (positively!); Will there be a C upgrade package? (probably not, tho this question is the one of the most common at fests); Is 512K enough? (yes, for now anyway); Why doesn't Tandy promote the CoCo more? (up to marketing, and besides which, lets face it, there are other things they make more money on, so let's just use what we have and educate users where we can). What kind of problems do you have? (the IRQ wire hack would solve most of these); and gosh I wish we'd taken notes on everything! Among the people there who helped out (much thanks!) were Paul Ward, James Jones of MW, Gary Robinson of Tandy, Bob Santy of PCDOS/RSDOS xfr util fame, Kent Meyers, and Kevin Pease (who showed prototypes of his new 68070/hires-gfx OS-9 board set). We truly had most of the OS-9 know-how in that one room! As always, I talked about the need to join a club or get online for help. Last year at Dale's seminar I asked who was on a forum or BBS. Out of two hundred people, only 6 were. This year almost everyone was! Remarkable difference. Plus virtually everyone knew of and used Shell+ and GShell+. It seems quite a bit of the software uploaded to CIS makes its way across country in almost no time.

We had an hour allotted for





the seminar, but they ended up having to kick the whole place out after two hours (!) because the next seminar was due to start. Apparently, that was some kind of record. Heck, we were just having a great time, is all. People had a lot to say. It could've gone on all day long, I suspect. Anyway, back to the UG booth for several hours. After the fest closed, we caught the last ten minutes of Gary Robinson's seminar on submitting software to Tandy. I wish I knew who that was in the room with the tape recorder... I'd like to get a transcript of this one!I do remember one thing: someone asked why Tandy doesn't sell developer's paks "like Apple" and Atari etc do. I had to jump in and say something myself on that one... the devpaks from those guys sell for $350-$1000 and include stuff like tech info, schematics, and OS calls. "Gee, we get the OS calls in our L-II manual, and schematics for any Tandy hardware are easily available to anyone for dirt cheap", is what I said.The attendees were hugely pleased that at last Tandy had sent a rep to the Chicago fest... and they made a point of saying how much they appreciated Gary showing up.

Saturday night, a bunch of us got together for dinner (and later of course ended up in Kent's room for more talk. I'm always invited to the Falsoft party, but somehow never have gone to it over the years; instead I like to get together with the forum folk). There were a dozen of us, and we told the waitress that Dale Puckett was our "Dad". To our delight, she treated him just so, all the rest of the evening <hehe>... even tho half our group had grey hair and was obviously older than he was. If that's possible <grin>.

-o-

On Sunday I finally found time to get away from the booth for a few minutes.Mike Knudsen's UltiMuse MIDI software under OS-9 was very impressive.Everything is menu-driven, using the mouse, altho he also provides keyboard shortcut keys. The surprise of the show (to me) was Chris Burke's 256K ram upgrade... it can also be upgraded to 512K... but provides a way to start out cheaper. About $90. Owlware had "WindowWriter", a mouse-driven text-screen word processor... several people (including Kent) bought a copy, so hopefully we'll have some reviews soon. It has two cursor blocks...mouse and text, with cut/paste and a kind of clipboard.

Just before the fest shut down at 3pm Sunday, they took the usual Rainbowfest group picture. In honor of Gary Robinson being at his first fest, I joined him towards the front of the group... so you might actually get to see what some of us look like finally.

That's about it. By 4pm the place was deserted. We left Monday at noon. Over the weekend, I had sold about 15 copies of my book, which was enough for Marsha and I to make it back to North Carolina again (whew!).

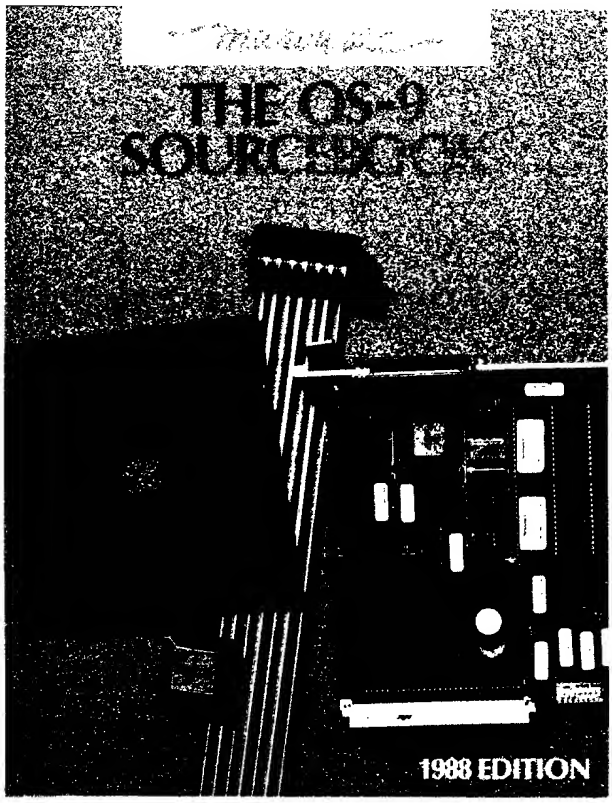
-o-

Friends we saw there included: Simmy Turner, Rod Motto, Dave Hansen,Roger Krupsky, Larry Olson, Mike Knudsen, Eric Crichlow, Tim Wilhite,Doug Dalton, Scott Griepentrog, Chris Burke, Dick White, Paul Ward, James Jones, Steve Blasingame, Dave Philipsen, Marty Goodman, Kevin Pease, George Dorner, Larry? Strong, Rick Adams, Ron and Tracy Lammardo, Mark Griffith, Bob Puppo, and Bruce Isted couldn't make it... and we missed them!

Some of the show specials that I remember included monitor stands for $1, CoCo-3's for $100, Orch-90 for $20, MultiVue for $25, L-II for $59. Also one-button mice were $10, two-button $39, Magnavox monitors for $250, ST251 40-meg bare drives for $299, complete 5-meg (no case) HD systems for about $125, and lots of deals on floppy drives of all types.

LIST OF COMPANIES AT THE FEST

Alpha Software - OS-9 utilities
AZ Small Computers - hard disk
Burke&Burke - OS-9 utils, hard disks, RAM upgrades
Cer-Comp - rsdos utils, peripherals
CBUG - floppy disks and accessories
CoCo Connection - rsdos utils
Computer Plus - Tandy hardware
Delphi - network
DISTO/CRC - peripherals
Gamepoint - rsdos games, Rascan video digitizer
Gimmesoft - games
Glenside CC club - buttons, t-shirts
Howard Medical - monitors, hard disks
Microcom - rsdos/OS-9 utils, peripherals
Oblique Triad - adventure games
Owl-Ware - hard drives, Window Writer/OS-9
Orion Tech - telecom soft/hardware
Radio Shack - Tandy soft/hardware
Rulaford Research - Lyra, MIDI synths
Second City - OS-9/rsdos software, UltiMuse MIDI
SpectroSystems - ADOS
SRB Software - games
StG Computers - OS-9 utils, bbs, printer dumps
Sugar Software - OS-9 calligrapher, educational software
Sundog Systems - games
Zebra Systems - peripherals,



Editors Note: Since Kevin mentioned the MW Sourcebook & Catalog, I thought I'd include some pics... and the MW addresses, now you have no excuse! (grin).

Microware Systems Corporation

Corporate Headquarters
1900 N. W. 114th Street
Des Moines, Iowa 50322
Phone: 515/224-1929
Fax: 515/224-1352

Western Regional Office
4401 Great American Parkway
Santa Clara, California 95054
Phone: 408/980-0201
Fax: 408/980-1671

Microware Japan Ltd.
41-19 Honcho 4-Chome
Funabashi City
Chiba 273, Japan
Phone: 0474(22)1747

RAVE Real Time Audio/Video Environment

ager (GFM) provides the audio, video and input drivers needed to support the run-time user interface. The input divers support a keyboard, as well as pointing devices. The keyboard driver supports any keyboard, from an IBM PC compatible keyboard, to a custom keyboard designed specifically for an application. The pointing device may he a mouse, touch pad, touch screen, or any device that returns X:Y coordinate information.

The Graphics Support Library (GSL) builds upon the GFM to create the more complex concepts required for an application. These include controls, indicators, and menus. Controls are objects on the display that mimic the behavior of switches. Indicators are objects on the display that mimic the behavior of output devices. Both controls and indicators may be implemented as computer generated objects, or by digitizing an actual image of the device.

RAVE's third package, known as the Presentation Editor, draws upon the GSL and GFM to provide an interactive, menu-driven development environment for building applications. With the Presentation Editor, designers create the controls, indicators, and menus supported by the Graphics Support Library.

Designers interact with the Presentation Editor via a keyboard and mouse. Audio can be input directly through a microphone, or loaded from disk. Video can either be captured via a camera, or built from scratch using graphics primitives. A Paintbox can be used to modify either computer-generated or real-world video images.

To use RAVE on new hardware, designers need only to port the low-level graphics, audio, and input drivers.

And RAVE was implemented as an extension to Microware's OS-9 Real-Time Operating system. RAVE provides users with complete access to the real-time functionality of OS-9, which is vital for many process control applications. Exceptionally fast interrupt response, preemptive task switching and a ROMable modular architecture make OS-9 the operating system of choice for real-time factory floor applications.

4.0 OS-9: Advanced Real-Time Operating System.

OS-9 has emerged as the pre-eminent real-time operating system for the Motorola 68000 family of microprocessors. OS-9 incorporates a number of powerful programing features such as multitasking, "SHELL" user interface, utility programs, hierarchical file structure and record locking, a complete suite of resident development tools, networking and a host of high-level languages including a C Compiler with UNIX/BSD 4.2 extensions. The versatility of OS-9 in being both a real-time kernel and a real-time operating system has made OS-9 the leading choice for process control, data acquisition, communications and robotics systems world-wide.

OS-9 includes a full-function real-time kernel and independent file managers to accommodate virtually every class of I/O device, peripherals, networking and interprocess communications. OS-9 takes a modular approach to system configuration to develop a complete computer system. These modules and associated file managers include the OS-9 kernel, math libraries, C libraries, the system security module, sequential character file manager, random block file manager, sequential block file manager, network file manager and the Berkeley socket file manager. Consequently, system integrators need only integrate the modules that are required for each application. This significantly decreases the memory size required for each system while increasing system performance. Also, the modular structure of OS-9 allows any system to be easily expanded.

Microware authors all of its compilers and interpreters. These include: C, Fortran, Pascal and Basic. These compilers are optimized for OS-9 and produce compact, position independent, re-entrant, ROMable code required for advanced real-time applications. As an added benefit, users of OS-9 have the ability to edit, compile and debug directly on the target system which significantly decreases the time necessary for system integration. Microware is a leader in C compiler technology for real-time environments. It's C compiler architecture and libraries are modeled on the proposed ANSI C standard and conform to the Kernighan & Ritchie standard.

5.0 The Graphics File Manager

The Graphics File Manager (GFM) brings together all the physical resources needed for the user interface. This includes a video driver, an audio driver, and drivers for the input devices. The diagram below shows the hierarchy supported by the GFM.

The Video Driver provides a very robust set of drawing and block copy functions. An application may draw lines, rectangles, polygons, circles, ellipsis, rectangles with rounded corners, or text. With all drawing functions, the application may select several variations such as drawing with a pattern, using variable size pens, using dashed lines, or any combination of the above.

All drawing is done on what is called a drawmap. The application may create several drawmaps, and select any one to be displayed. The application may select a new draw map to be displayed at any time.

Simple and complex regions are also supported. These regions may be created and mixed very easily. After creating a region, it may be drawn onto a drawmap, or used for clipping. In the later case, all drawing outside the region will be clipped (not drawn).

A very powerful set of block copy and exchange functions are also supported This allows an application to copy or exchange images between drawmaps. Part of an image can be declared as transparent. This is done by using a clipped region, or using a transparent color.

The Audio Driver supports playback of pre-recorded audio. This audio data can be in memory, or loaded off of disk. The quality of the audio is only limited by the hardware.

Two types of Input Drivers are supported: pointer and keyboard. The pointing device may be a mouse, touch pad, touch screen or any hardware device that returns X:Y coordinate information. Any type of keyboard may be supported using the GFM. The keyboard may be PC compatible or one developed specifically for an application.

The following is a list of functions supported by the GFM:

Graphics Cursor Functions:

```
gc_col() Set cursor color
gc_hide() Hide cursor
gc_org() Set cursor origin
gc_pos() Set cursor position
gc_ptn() Set cursor pattern
gc_show() Show cursor
```

Drawmap Control Functions:

```
dm_close() Close drawmap
dm_copy()  Drawmap copy
dm_creat() Create drawmap and
return ID
dm_create() Create drawmap and
return descriptor
dm_dup() Duplicate drawmap and
return ID
dm_dupe() Duplicate drawmap
and return descriptor
dm_exch() Drawmap exchange
dm_irwr()  Irregular write to
drawmap
dm_org()  Set drawing origin
dm_rdpix()  Read single pixel
from drawmap
dm_read()  Read from drawmap
dm_tcopy() Drawmap copy with
transparency check
dm_texch() Drawmap exchange
with transparency check
dm_write()  Write to drawmap
dm_wrpix() Writes single pixel
to drawmap
```

Display Control Functions:

```
dc_daplsiz() Return display
size and resolution
dc_getclut()  Get a single
CLUT value
dc_getcluts() Get a range of
CLUT value
dc_relea() Release pending
signal request
dc_setclut() Set a single CLUT
value
dc_setcluts() Set a range of
CLUT values
dc_ssig()  Send signal on
video interrupt
```

Region Functions:

```
rg_creat() Create region
rg_del() Delete region
rg_diff() Create region from
difference of two regions
rg_isect() Create region from
intersection of two regions
rg_move() Move region
rg_union() Create region from
union of two regions
rg_xor() Create region from
exclusive or of two regions
```

Drawing Parameter Functions:

```
dp_afnt() Activate font
dp_clip() Set clipping region
dp_dfnt() Deactivate font
dp_gfnt() Get font
dp_paln() Set pattern align-
ment
dp_pnsz() Set pen size
dp_pstyl() Set pen style
dp_ptn() Set drawing pattern
dp_rfnt() Release font
dp_scmm() Set character code
mapping method
dp_scr() Set color register
dp_tcol() Set transparency
color
```

Video Inquiry Functions:

```
viq_txtl() Calculate length of
text
viq_cpos() Return relative
character positions
viq_icps() Return character
positions for justified text
viq_fdta() Return font data
viq_gdta() Return glyph data
viq_pntr() Test if point is in
region
viq_rloc() Return region
location
viq_rinfo() Return region
```

descriptor information
viq_dminfo() Return drawmap descriptor information

Graphics Drawing Functions:

dr_bfil() Fill a bounded area with a pattern
dr_carc() Draw a circular arc
dr_circ() Draw a circle
dr_copy() Drawmap copy with clipping
dr_cwdg() Draw a circular wedge
dr_dot() Draw a dot
dr_drgn() Draw a region
dr_earc() Draw an elliptical arc
dr_elps() Draw an ellipse
dr_erect() Draw an elliptical cornered rectangle
dr_ewdg()Draw an elliptical wedge
dr_ffil() Flood fill an area
dr_jtxt() Output justified text
dr_line() Draw a line
dr_pgon() Draw a polygon
dr_plin() Draw a polyline
dr_rect() Draw a rectangle
dr_text() Output graphics text

Standard Character Output Functions:

co_afnt() Activate font
co_cod() Set character output drawmap
co_dfnt() Deactivate font
co_scmm() Set character code mapping method
crt_cup() Cursor up
crt_cdwn() Cursor down
crf_crght() Cursor right
crf_clft() Cursor left
crf_home() Cursor home
crt_cr() Carriage return
crt_curyy() Cursor address (X:Y location)
crt_dlin() Delete line
crt_ilin() Insert line
crt_shwcur()Show cursor
crt_hidcur() Hide cursor
crt_ceol() Clear to end-of-line
crt_ceos() Clear to end-of-screen
crt_revon()Start reverse video
crt_revoff()End reverse video
crt_ulon() Start underlining
crt_uloff() End underlining
crt_cscrn()Clear screen
crt_ichar() Insert character
crt_dchar()Delete character
crt_wrapon()Turn on auto-wrap mode
crt_wrapoff() Turn off auto-wrap mode

Pointer Input Functions:

pt_coord()Return pointer coordinates
pt_org() Set pointer orgin
pt_pos() Set pointer position
pt_rel() Release pending signal request
pt_ssig() Send signal on pointer change

Keyboard Input Functions:

kb_rdy() Check for keyboard data ready
kb_rel() Release pending signal request
kb_ssig() Send signal on data ready

Screen Management Functions:

sn_act() Activate screen
sn_alink() Link action region to screen
sn_close() Close screen
sn_cp() Call cursor process
sn_cpact() Activate cursor process for a screen
sn_cpdeact() Deactivate cursor process for a screen
sn_deact() Deactivate screen
sn_dmlink() Link drawmap to a screen
sn_getact() Return active screen
sn_info() Return screen information
sn_lower() Lower screen in stack
sn_open()open screen
sn_raise() Raise screen to top of stack

Action Region Management Routines

ar_absxy()Return absolute co-ordinates
ar_act() Activate action region
ar_close()Close action region
ar_closesub()Close sub-action regions
ar_deact() Deactivate action region
ar_find() Find action region containing coordinate
ar_grab() Start synchronous grabbing
ar_info() Return action region descriptor
ar_lower() Lower action region in stack
ar_mask() Mask incoming messages for action region
ar_move() Move action region
ar_open() Open action region
ar_raise() Raise action region to top of stack
ar_redefine() Redefine action region
ar_relxy() Return relative co-ordinates
ar_ungrab() Terminate synchronous grabbing

Message Management Functions:

ms_flush() Flush message queue
ms_journ() Turn on/off journal mechanism
ms_read() Read message queue
ms_ready() Check message queue
ms_release() Release message queue
ms_signal()Send a signal when a message is available
ms_stat() Return pointer status
ms_unread()Write a message to head of queue
ms_write()Write message to end of queue

Action Cursor Functions:

ac_act() Activate graphics cursor
ac_col() Set graphics cursor color
ac_deact() Deactivate graphics cursor
ac_org() Set graphics cursor and pointer origin
ac_pos() Set graphics cursor position
ac_ptn() Set graphics cursor pattern
ac_track()Turn on/off automatic tracking of graphics cursor

Soundmap Functions:

sm_close()Close soundmap
sm_creat() Create a soundmap
sm_create()Create a soundmap
sm_in() Input to a soundmap
sm_into() Return pointer to soundmap descriptor
sm_loop()Set soundmap loopback point
sm_ off() Turn off soundmap option
sm_out() Output a soundmap option
sm_ready() Check audio queue

6.0 The Graphics Support Library

The Graphics Support Library (GSL) builds upon the GFM to create the more complex concepts needed by an application. These includes controls, menus and indicators.

Controls are objects on the display that mimic the behavior of switches and slide- bars. The user may interact with a control on the display to turn something on or off. A slide-bar could be used to mimic something like a volume control.

Menus (called requests in the GSL) give designers a means of configuring an interface that enables users to select various controls and indicators. These menus may be simple text or complex images. The application may also select whether the menu is displayed all the time (menu bar), or only when appropriate (pop-up menu).

Indicators are objects on the display that mimic the behavior of output devices. These include a digital readout, level indicator, LED meter, linear, meter, and a strip chart recorder. These may be computer-generated objects or actual images of a real device.

In addition to the on screen objects supported by the GSL, there are many housekeeping functions supplied that make the task of writing an application a bit easier. These include message handling, a clip board and resources.

Following is a list of functions supported by the GSL:

Resource Manager Functions:

res_copy()Copy a resource
res_count()Find resource type count
res_current()Get current resource module
res_free() Free resource module
res_get() Get a sharable resource
res_id() Find item
res_itype()Find resource type
res_load()Load or link to a resource module
res_loadc()Load or link to a country-dependent resource module
res_set() Set current resource module
res_share() Determine if resource is sharable
res_size() Get size of a resource
res_tycount() Find item count for a type

Request Manager Functions:

req_activate() Activate a modeless request
req_create() Create a request
req_deactivate() Deactivate a modeless request
req_def() Execute standard definition function
req_draw() Draw a request
req_free() Free a request
req_handler() Execute the request's message handler
req_hide()Hide a request
req_itemmark() Change an item's checkmark
req_itemstate() Change an item's state
req_make() Make a modal request
req_move() Move a request
req_setcurs() Place the pointer in a request item
req_show() Show a modeless request

Control Manager Functions:

cntl_action()Execute application action
cntl_activate() Activate a control
cntl_bhv() Execute behavior function
cntl_create()Create a control
cntl_deactivate() Deactivate a control
cntl_def() Execute definition function
cntl_delete() Delete a control
cntl_dolist() Perform a function on a list of controls
cntl_draw() Draw a control
cntl_handler()Execute message handler
cntl_hide() Hide a control
cntl_max()Set the maximum value of a control
cntl_min() Set the minimum value of a control
cntl_move() Move a control
cntl_setcurs() Places the pointer on a control
cntl_show()Show a control
cntl_state()Set the state of a control
cntl_value()Set the value of a control

Indicator Manager Functions:

ind_create() Create an indicator
ind_def() Call an entry point in a definition function
ind_delete() Delete an indicator
ind_draw() Draw the entire indicator
ind_show() Show an indicator
ind_hide() Hide an indicator
ind_value() Set value of an indicator

Clipboard Manager Functions:

clip_close() Close the clipboard
clip_count() Get clipboard type count
clip_counter() Get clipboard counter
clip_getptr() Get pointer to clipboard data
clip_link() Link to the Clipboard
clip_load() Load the clipboard
clip_owner() Get pointer to name of clipboard owner program
clip_read() Read the clipboard
clip_ready() Get data byte count
clip_rewrite() Re-write in the clipboard
clip_ropen() Open for read access
clip_save() Save the clipboard
clip_type() Get clipboard type
clip_unlink() Unlink from the clipboard
clip_wopen() Open for write access
clip_write() Write in the clipboard

Dispatch Manager Functions:

hndlr_create() Create message handler
hndlr_delall() Delete all handlers from an action region
hndlr_delete() Delete message handler
hndlr_dsptch() Dispatch a message
hndlr_prab() Set up message

grabbing
hndlr_mask() Change the message mask for a handler

Overlay Window Manager Functions:

ovly_close() Close an overlay window
ovly_open() Open an overlay window
frm_dbox() Create dialog box frame
frm_oline() Create outline frame
frm_sbox() Create shadowed box frame
dbox_open( ) Create a dialog box overlay window
obox_open() Create a outlined box ovelay window
sbox_open() Create a shadowed box ovelay window

User Preference Functions:

pref_get() Get preference
pref_link() Link to preference
pref_read() Read preference from a file
pref_set() Set preferences
pref_write() Write preferences to a file

Internationalization Functions:

intl_currency() Create a currency string
intl_date() Create a date string
intl_free() Unlink the international data module
intl_ismetric() Check for Metric/English measurement system
intl_set() Set the international data module
intl_time() Create a time string
Association Table Functions:
atbl_create() Create an association table
atbl_dealloc() Deallocate an association table
atbl_delassoc() Delete an association
atbl_lookup() Look up an association
atbl_makeassoc() Make an association

Fast Block-Move Functions:

mv_bytes() Fast move bytes
mv_words() Fast move words

7.0 The Presentation Editor

The RAVE Presentation Editor is used to develop applications. The Presentation Editor automatically generates C source code for all the software routines necessary to control the user interface. The designer is not required to Write one line of A/V control code. The entire interface may be designed interactively by using this tool.

With the Presentation Editor, a designer can create the controls, menus and indicators supported by the GSL. These objects may be combined onto the display to form the interface for an application. Since RAVE supports real audio and real images in addition to those computer generated, the Presentation Editor has tools for manipulating this data.

The audio tools allow the designer to capture, edit, and playback audio segments. The video tools allow the same facilities for video images. In addition, a paint facility is available for creating new images or touching up images captured by camera or other means.

8.0 Configuring RAVE

RAVE is configured to provide both run-time and development environments.

RUN-TIME ENVIRONMENT

Rave's Graphic File Manager (GFM) and Graphics Support Library (GSL) form the foundation of the OS-9/RAVE run-time environment. This run-time environment provides a sophisticated man/machine interface for virtually any type of audio and video hardware; including mouse, touch screen, terminal, microphone and speaker. Furthermore, the GSL provides basic indicators and controls so software engineers can build an application without the use of more complex and expensive development tools.

The entire RAVE run-time environment is completely ROMable, compact in size and designed expressly for use in embedded applications. An OS-9/RAVE application can be embedded in systems supporting as little as 128K of system RAM.

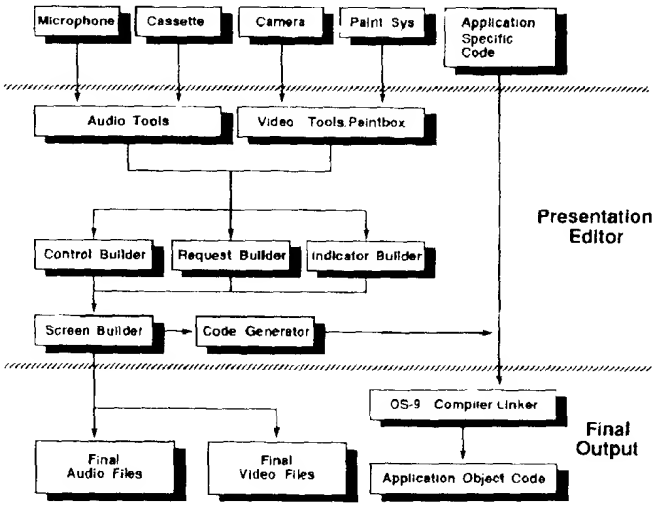
The RAVE run-time environment is configured for distribution under Installation Pak and OEM License. These packages include all of the software source code and documentation necessary to install the run-time environment on a given hardware platform.

DEVELOPMENT SYSTEM

The Presentation Editor and its related hardware (microphone, frame grabber, camera, paint box, video monitor, etc.) comprise the RAVE development environment.

Microware understands that the ever increasing pace of technological advancements make product design time a key factor of a product's commercial success. The time devoted to developing a man/machine interface using conventional programing techniques can render a product obsolete long before it is brought to market. The Presentation Editor reduces the amount of time spent on developing the application by automatically generating, in C source code, all the software routines necessary to control the user interface. Simply put, by using the RAVE development environment, you can get your product to market quicker with less expense.

FLOW CHART FOR BUILDING A RAVE APPLICATION



RAVE Real Time Audio Video Environment

The Users Group....

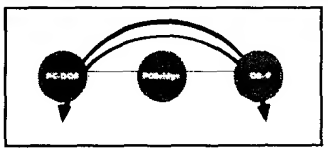
The OS-9 Users Group is an international non-profit organization of approximately 800 members (and growing) devoted to exchanging and distributing information about, and public domain software for, all available versions of the OS-9 Operating system. The OS-9 Users Group is the only independent group officially recognized by Microware (the developers of OS-9) as an official voice of its users.

The OS-9 Users Group periodically publishes this newsletter entitled **"MOTD"** which contains many useful articles, software listings, and other information helpful in keeping OS-9 computing enjoyable and rewarding. Other membership benefits include free technical help referrals (by mail or electronic BBS) and significant discounts on the purchase of individual volumes of the OS-9 Users Group Public Domain Software Library. One year memberships in the group cost $25.00 for individuals and $150 for companies (corporate membership) and includes a subscription to the MOTD newsletter, one free disk of public domain software (archive set of entire Library for corporate members), and the right to purchase additional disks of software at a very reasonable cost. The group's public domain software library currently has over 56 individual volumes of software comprised of almost 300 individual programs. The library is constantly growing due to the group's policy of sending one volume (disk) from the library free for each individual program donated by a member. (note, although UG software is available from other sources, only MEMBERS receive the latest, and librarian maintained versions).

To join the OS-9 Users Group, fill out the application form reproduced on page 9 (or facsimile thereof) and send to the UG Tampa address.

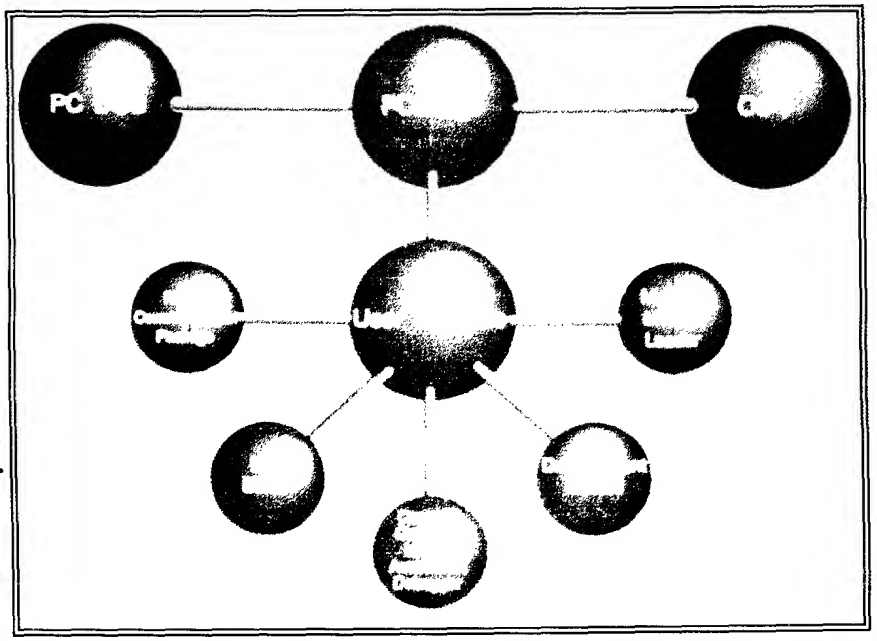
After you join, you will receive a copy of the current issue of the OS-9 Users Group newsletter ("MOTD"), and soon after that, the "starter" diskette, UG Library Volume #0, with software of the type useful in getting you started with both OS9 and the Users Group. Current members who renew their membership will receive a UG "donation credit" post card, which may be redeemed for most UG products and services at any time during your membership.

Product Overview: PCBridge



PCBridge is a low-cost, easy-to-use, PC-hosted development and supervisory system for intelligent 680x0 based applications.

PCBridge can be linked to a variety of real-time tasks, and can provide a gateway for OS-9 data and applications on PC-DOS. A total distributed application can be developed on the PC host and integrated into an OS-9 real-time environment using PCBridge.



PCBridge resides on the PC host system, with a special utility package resident on the OS-9 system. Using PC's, XT's or AT's as a front end, PCBridge links to the OS-9 target system via a serial line. Applications are developed on the PC and then downloaded to the target for testing or execution. The OS-9 target system can be configured as a ROM-based system with limited RAM or as a complete professional development environment.

The user interface appears as a pop-up menu. This allows the user to select a function (edit, compile, transmit, etc.), indicate the necessary parameters and PCBridge performs the operation without further user intervention. PCBridge menu selections are made using key words, keypad cursor keys, or mouse.

PCBridge Environment

PCBridge provides a platform that allows users to share C and Assembler applications and data between PC-DOS and OS-9 through an RS-232 protocol link. PCBridge users have access to the OS-9 operating system allowing real-time process execution, data transfer, monitoring and debugging from a PC-host terminal.

Microware's PCBridge is distributed on IBM formatted 3 1/2" or 5 1/2", diskettes. Also included is an OS-9 format diskette of OS-9 Utilities for installation on the OS-9 target system.

PC-Host Hardware and Software Requirements

PCBridge operates on the following PC-DOS systems:

- IBM PS/2
- IBM PC/XT/AT
- Any IBM PC/XT/AT Compatible System

A minimum of 512 Kbyte RAM and 1 Mbyte of on-line disk storage is required for operation. The PC must be running PC-DOS Version 3.2 or greater and be equipped with an available serial port

OS-9 Hardware and Software Requirements

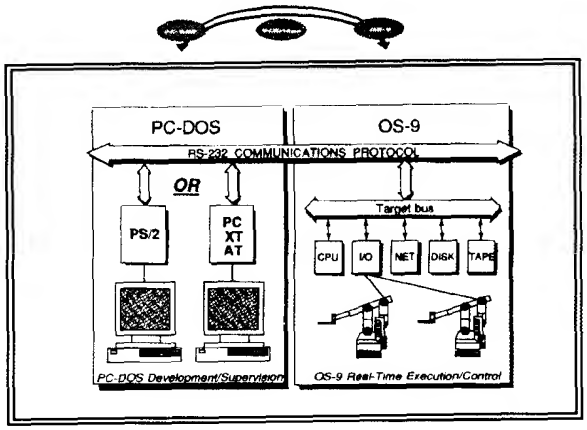
PCBridge operates on any OS-9/680x0 system that supports Version 2.0 (or later) of the operating system.

Initial installation of PCBridge requires either Industrial OS-9, Personal OS-9 or Professional OS-9 operating on the target system. All OS-9 resident modules of PCBridge consist Of standard OS-9 memory modules and can be ROMed in any Industrial OS-9 system after development and system reconfiguration.

PCBridge XCC Cross C Compiler

The PCBridge XCC Cross C Compiler is a **full implementation of Microware's OS-9 C Compiler** and contains all executable, library and definition files contained in the OS-9 resident version. OS-9/XCC was expressly designed for the 68000 family instruction set and includes complete 68881 floating point coprocessor support. The compiler performs multilevel optimization in order to produce extremely fast, compact object code. Compiled programs are completely ROMable, reentrant and position independent.

When transferred from the PC-DOS host to the target OS-9 system, C object code can take full advantage of OS-9's shareable memory modules which contain standard library I/O and mathematical routines. OS-9/XCC can automatically generate code which utilizes these system wide modules rather than linking redundant copies of the standard library to every executable program. **By taking advantage of OS-9's unique modular environment in this way, programs require less memory, less disk space, load faster and can be updated externally by a host system.**



Each PCBridge package contains the following modules:

- **OS-9/XCC C Compiler**
  Assembler
  Linker
- **User State Debugger**
- **OS-9/RS-232 Communications Package**
- **OS-9/Development Utilities**
- **Memory Module Loader**

Review: Basic09 XREF by Bill Brady

Every once in a while you discover a sleeper. That is: a program that has been around for a while that you wish you'd had a long time ago. That happened for me with XREF.

Now that Basic09 is supplied with OS-9 Level 2, more and more users are discovering it's power. As you may know, Wiz and Wiz Professional terminal programs are written in Basic09. I believe that Window Writer is also done in B09. So commercial and shareware Basic09 programs are coming to the fore.

You may not know what a Cross-Reference program does. It's a fairly simple thing, but you must know everything about the compiler to write one, and thoroughness is an absolute must. XREF eats Basic09 source files, and outputs a listing... both of the source, and, at the end, a cross reference of every variable and line number contained within the program.

The normal Basic09 list function puts a raw byte count in front of each line. But XREF assigns simple, but understandable line numbers. (not Basic09 line numbers.. just the number of that line in the listing).

The 'manual' is a formatted printout on a dot matrix type printer. You normally don't get a manual at all with a utility. It provides instructions on how to set up the program, which although are not step by step, are more than adequate for an experienced Basic09 programmer.

Using XREF you can easily find out where any particular Basic line number is referenced. More importantly for me, however, is that you can quickly spot unnecessary line numbers.

What is an 'unused line #? Why is it necessary to spot them? Well, it depends on your programming style. In my case, when I am debugging code, I often stick in extra line numbers for diagnostic purposes. Later, when the code is fixed, that line number is no longer referred to elsewhere in the code. I really should go back and 'clean' up right away, but I am usually off in pursuit of the next bug instead.

With XREF, I can go back later and easily spot any of these left overs because they will appear in the XREF with only one XREF line number opposite... the one in which they occur.

In addition to Basic line numbers, XREF also does variables. Again, you can use the XREF to find 'loners' in this category. Great for reducing run time memory requirements.

XREF comes in two versions: with or without source. The Basic09 program 'XREF' is a short 'front end' that calls XRMain and GetWord, and in either case is supplied in source form. Why? so you can customize... set your output path, and pathways to the ancillary file 'KEYWORDS' (like /DD/SYS/KEYWORDS.

I gave XREF a quick try under OSK, and it loaded just fine. I don't see any reason why it wouldn't work with some modifications, you'd want the full source version of course.

XREF's listing option is supplied as a separate program. It yields a formatted listing without the cross-reference... you use this version for speedy pretty program listings only.

Both programs are supplied in two versions, one for older printers which don't support the FF, (form feed or top-of-form), character. I use an Epson with a sheet feeder, and performance was flawless for me, although I did have to set the printer for 12 cpi to keep it from line wrapping without the programs knowledge.

XREF has been around a while, but then so has Basic09, and XREF is right in sync today. I am not sure of current pricing, but you can pick yourself up a copy of 68 Micro.

Bacic09 XREF is a must have program for anyone serious about Basic09. I know of no subsitute for it.

Basic09 Cross Reference Utility (XREF)

Applied Computer Technology, Inc
6435 Summer Ave.
Memphis TN 38134

(901) 377-3503

From the Editor - continued.....

to a different layout for the MOTD. I am using different fonts: Bookman and **Cooper Black.** The point size is smaller, and there are now 4 columns, but there is also more 'white space'. The idea is to get more info in less space, and if the new printer is a good as I think he is, you will find this MOTD very readable.

I have also purchased a better scanner: the AppleScan. It provides 300 dpi output and does 4 bit grayscale. I apologize for those who found the template in the last issue un-readable; I will re-scan with the new scanner if you like. (It was partly the printers fault). The AppleScan also works better with my OCR software. This is how the RAVE, PCBridge, and OS-9 Signals segments got in this issue. Let me know if you want more.

Speaking of 'letting me know, letters to the editor have dropped off somewhat. Are you guys so happy with the state of affairs that you no longer want to be heard?

Although Kevin does not have an article here, he has asked me to stay on as Editor. I am not sure who the new librarian will be, nor correspondence secretary or Directors-at-Large.

I'd like to say a few words about reviews. We have received products that we have attempted to review but could not because they don't work, or don't work without modification of OS-9. Rather than simply state that fact, we attempt to work with the producer to make that product useful. This usually works, but not always. Keep in touch, because soon I'll need some reviewers. When you write, you must let me know all that you have, including software!

Include your phone number on all correspondence!

My HCA/WD system has been down. Not because of any defect in the basic product, but because the Western Digital controller board I used had a bad solder job. (& one chip improperly inserted. At this point, I can confirm that the HCA/WD and WDDisk version 5 driver works fine with Bob Santys PCDOS.

You may have also noticed that the OS-9Boots section is missing. Have we covered all of the new OS-9 vendors? If you know of someone who is just getting into the OS-9 business, give me a holler. KEN-TON, in particular has indicated that his Boots ad was very successful

We need reviewers for OSK software. But you must have a fairly extensive hardware set. I need someone to review cross-compilers.

We need some more User written articles.

OS-9!

SUBMISSIONS

Articles, letters and advertisements will be accepted in the following formats:

VEF, GIF, MACPAINT, MACDRAW, CANVAS, TIFF, PICT, ThunderScan, MS WORD or WORKS, MACDRAFT, READY SET GO!, or Plain text files, ON ANY OF THE FOLLOWING: 5.25" ALL OS-9 FORMATS, 3.5" COCO-ATARI, 3.5" MAC 400K OR 800K, OR VIA E-MAIL TO THE EDITOR ON GENIE OR DELPHI. You can upload to my mainframe, if it is on line, between 8am and 6pm EST User:Guest Password: CIVIL, 1200 baud. Call voice after 6pm. The number is 301 952-1761.

NO PAGEMAKER, Post Script, or CIS mail please.

Please include complete address, user #, and phone number on all submissions. Also tell me what you want us to do with whatever you are sending, Article, Ad, or Letter to the Editor, etc.

Include your phone number on all correspondence!

OS-9 Users Group Software Library Order Form

(For Members of the OS-9 Users Group ONLY)

FROM... Member Number: [ ]

Name: _____

Address: _____

City, State, Zip: _____

Telephone Number(s): _____

Computer Disk format Type (A, B, C or D - from instructions):
Format Code(s) of floppy disks you can read (1, 2, 3, 4, 5 and/or 6 - from instructions):

		Format:		Price:	
Quantity:	**Volume # ****	**Type:**	**Code****	**Each****	**Total:**

Total amount for Individual Disks:

*** Note: Information indicated by a "**" above should be taken from the current UG Library Volumes listing, as reproduced in the UG newsletter."*

Order Archive Set of Entire UG Library below (available as multiple set of disks in format code #6 ONLY):

		Format:		Price:	
Quantity:	**Volume #:**	**Type:**	**Code:**	**Each:**	**Total:**
	Archive	**A**	**6**	**$100**	
	Archive	**B**	**6**	**$120**	
	Archive	**C**	**6**	**$130**	
	Archive	**D**	**6**	**$110**	

Total amount for Archive Sets: _____

+ Amount from above: _____

Total Payment enclosed: _____

Please write us if you desperately need any of the above disks in any format other than the ones we currently offer (such as 8" disks, other formats, etc.). We may be able to put you in touch with another volunteer UG member who could do the conversion for you.

Master Card # [ ] Expiration Date [/]

Visa Card # [ ] Expiration Date [/]

Signature: Todays Date [/ /]

Make all checks payable to "OS-9 Users Group"
Send disk order to: OS-9 Users Group

ATTN: DISK ORDER

1715 East Fowler Ave., Suite R237
Tampa, FL 33612

Please allow 6 weeks before inquiring about your order.

Application for New or Renewal Membership in the OS-9 Users Group

RENEWAL membership, your membership number please: [ ]

Type of membership applied for: [ ] Corporate [ ] Individual

Name: _____

Company Name (for Corporate memberships only):

_____

Address: _____

City, State, Zip: _____

Country: _____

Telephone Number(s): _____

Type of Computer System you are running OS-9 on:

Format of OS-9 floppy disks you can read (please check all that apply):

5.25" [ ] 8" [ ] 3.5" [ ] Single-Sided [ ] Double-Sided [ ]

Single-Density [ ] Double-Density [ ] Hi Density [ ]

Number of tracks per side (35, 40, 80, etc.) [ ]

FORMAT: Microware [ ] CoCo [ ] Atari ST [ ] Mizar [ ]

Other OS-9 Format (please describe):

ELECTRONIC ADDRESS(S): CompuServe #: [ ]

Delphi : [ ] GEnie: [ ]

BIX : [ ] UseNet : [ ]

Would you be willing to volunteer your time and effort to work on one of the OS-9 Users Group's committees or run for an office? If so, please describe the capacity in which you would like to get involved:

[ ] Check here if you DO NOT want your name and address printed in a member directory.

[ ] Check here if you DO NOT want your telephone number given out to other OS-9 Users.

Number of years membership you are enclosing payment for: [ ]

INDIVIDUAL MEMBERSHIPS IN THE OS-9 USERS GROUP ARE $25.00 PER YEAR, (overseas $30 US). CORPORATE (COMPANY) MEMBERSHIPS ARE $150, (overseas $160 US). PER YEAR

(sorry, but the Library archive set is presently only available on 5.25" ds dd 80 track or 3.5" ds OS-9/ST format disks)

(The OS-9 Users Group is a non-profit organization, any other charitable donations will also be gladly accepted.)

Total Payment enclosed: $ [ ]

Method of payment: [ ] Check Number [ ]

Master Card # [ ] Expiration Date [/]

Visa Card # [ ] Expiration Date [/]

Signature: Date [/ /]

Make all checks payable to "OS-9 Users Group"

Send application to:
OS-9 Users Group

ATTN: MEMBERSHIP

1715 East Fowler Ave., Suite R237
Tampa, FL 33612

Please allow 4 to 6 weeks before inquiring about your application.

Instructions For Ordering UG Library Disks

PLEASE READ CAREFULLY BEFORE SENDING IN YOUR DISK ORDER!!

1) Note what type of computer system you a running OS-9 on. Theoretically, all OS-9 systems should be able to read one or more of the following format disks:

Type "A": Standard Microware Format (track 0 is single density and 10 sectors per track, all other tracks are 16 sectors long; 0 sector offset)
Type "B": Tandy Color Computer Format (all tracks are double density and 18 sectors long; 1 sector offset)
Type "C": Atari 3.5" Format (Micro-Floppy 3.5" disks, 80 tracks, all tracks double density and 16 sectors per track; 0 sector offset)
Type "D": OS-9 Users Group format (all tracks are double density and 16 sectors long; 0 sector offset) This format is sometimes referred to as "Mizar format".

2) Determine what type of floppy disk drives you have on your OS-9 computer system. The available choices are...

Code	Type "A" Std. OS-9 Format: (5.25")	Type "B" CoCo OS-9 Format: (5.25")
1	35 track ss sd	35 track ss dd
2	35 track ds sd	35 track ds dd
3	40 track ss sd	35 track ss dd
4	40 track ds sd	40 track ss dd
5	40 track ds dd	40 track ds dd
6	80 track ds dd	80 track ds dd

Code	Type "C" Atari OS-9 Format: (3.5")	Type "D" OS-9 UG / Mizar Format: (5.25")
1	80 track ss dd	40 track ss dd
2	80 track ss dd	40 track ss dd
3	80 track ss dd	40 track ss dd
4	80 track ss dd	40 track ss dd
5	80 track ss dd	40 track ds dd
6	80 track ds dd	80 track ds dd

KEY: ss=single sided, ds=double sided, sd=single density, dd=double density

For reference, it should be noted that some OS-9 UG media format codes equate to certain official Microware media format codes. In particular, the following equivalences are noted:

Media Format Codes

Microware Systems		OS-9 Users Group
5403	...is equivalent to...	A5
5803	...is equivalent to...	A6
3807	...is equivalent to...	C6
5407	...is equivalent to...	D5
5807	...is equivalent to...	D6

Also note that the following Microware standard disk formats are also available by special request. Please do not order them unless it is the ONLY format you can use on your computer:

Microware Systems		OS-9 Users Group
3803	...is equivalent to...	C6a
38W7	...is equivalent to...	C6b
58W7	...is equivalent to...	D6b

Choose the OS-9 Users Group format codes above which represent the formats of disks you know you are able to read. For example, most TRS-80 Color Computers can only read formats #B1, #B2, #B3 and (sometimes) #B4 above. This indicates that a person with a stock Color Computer should be careful to only order Library volumes with these format codes. DO NOT ORDER LIBRARY DISKS WITH FORMAT CODES OTHER THAN THE ONES WHICH YOU KNOW YOUR SYSTEM IS ABLE TO READ!

3) Choose the Volume Numbers of the individual disks you would like to order, carefully noting the format code of each volume you would like to order. IF THE FORMAT CODE OF THE DISK YOU WOULD LIKE TO ORDER DOES NOT MATCH THE CODE OF A FORMAT YOUR COMPUTER SYSTEM IS ABLE TO READ (as calculated in step 2 above), YOU SHOULD NOT ORDER THAT VOLUME!! Yes, this means that it is possible that a disk you want may not be available in a format you can read. This is sometimes necessary when an individual program on a disk is too big to fit on a small format disk.

In particular, it should be noted that Volumes #2, and #49 cannot be used on most TRS-80 Color Computers. Care should also be taken before ordering Volumes #6, #19, #39, #47, #55 and #56 (and any others whose format code is 4 or greater). Also note that the multi-disk "Archive Set" of the entire UG Library is only available on 80 track ds dd disks (format code 6) ONLY and can therefore not be used on a stock TRS-80 Color Computer.

4) Prices for individual volumes are as follows (as of April, 1988):

	Types "A","B" or "D" (5.25")	Type "C" (3.5")
1	$6.00 each	$8.00 each
2	$6.00 each	$8.00 each
3	$6.00 each	$8.00 each
4	$6.00 each	$8.00 each
5	$6.00 each	$8.00 each
6	$10.00 each	$10.00 each

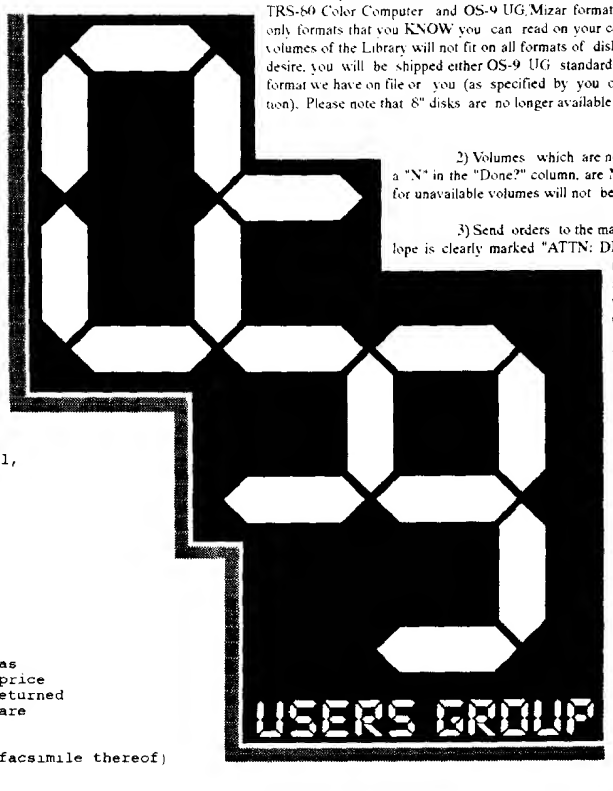
NOTE: PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE! Orders received 30 days or more after an official price change (as announced in the MOTD newsletter) are subject to the new price schedule. Orders sent with insufficient payment will be returned unprocessed unless specific instructions to do otherwise are included with the order.

5) Fill out the order blank reproduced in this issue (or facsimile thereof) with all of the necessary information.

OS-9 Users Group Software Library

Volumes - 8/88

No.	Done?	Title:		Format:
0.09	Y	New Member Intro		3*
1.01	Y	Spelling Checker (Improved; 6809 & 68K)		4
2.02	Y	Spelling Dictionary (102,681 words, 6809 & 68K)		5
3.01	Y	Word Processing Utils		1*
4.01	Y	Programming Utilities		1*
5.00	Y	File Processing Utils		1*
6.02	Y	Adventure Game (source)		4
7.02	Y	Adventure Game (object)		3
8.00	Y	General Interest (demo, games, finance)		1
9.00	Y	C Programmer's Tool Kit		3
10.00	Y	Math & Electronics		1
11.00	Y	Word Processing Utils	(disk #2)	1
12.00	Y	Programming Utilities	(disk #2)	1
13.00	Y	File Processing Utils	(disk #2)	1
14.03	Y	File Maintenance		1*
15.01	Y	Communication		1
16.00	Y	Hardware Customizations		1
17.00	Y	Basic09 Programmer's Tool Kit		1
18.00	Y	System Utilities		1
19.01	Y	Languages 1: XLisp (source)		4
20.00	Y	XLisp (object)		1
21.00	Y	File maintenance	(disk #2)	1*
22.00	Y	Programming Utilities	(disk #3)	1
23.00	Y	File Processing Utils	(disk #3)	1
24.01	Y	General Interest	(disk #2)	1
25.02	Y	Word Processing Utils	(disk #3)	1
26.01	Y	C Language Math Library (6809 only)		3
27.01	N	<undefined>		
28.00	Y	68K Utilities		1
29.00	Y	File Maintenance	(disk #3)	1
30.00	Y	File Processing Utils	(disk #4)	1
31.00	Y	Hardware Customizations	(disk #2)	1
32.00	Y	Hardware Customizations	(disk #3)	1
33.00	Y	System Utilities	(disk #2)	1
34.00	Y	Hardware Customizations	(disk #4)	1
35.00	Y	System Utilities	(disk #3)	1
36.00	Y	General Interest	(disk #3)	1
37.01	Y	Communication (6809 & 68K Kermit)	(disk #2)	2
38.00	Y	Programming Utilities	(disk #4)	1
39.00	Y	Communication (Freeware)	(disk #3)	4
40.00	Y	System Utilities	(disk #4)	1
41.00	Y	Programming Utilities	(disk #5)	1
42.00	Y	Coco Graphics		1
43.00	Y	System Utilities	(disk #5)	1
44.01	Y	Communication (Smod8)	(disk #4)	1
45.00	Y	Coco Graphics	(disk #2)	2
46.00	Y	Text Processing Utils (Sled)		2
47.00	Y	Text Processing Utils (68K Runoff)	(disk #2)	4
48.01	N	<undefined>		
49.00	Y	Text Processing Utils (MicroEMACS)	(disk #3)	5
50.00	Y	68K Utilities	(disk #2)	1
51.00	Y	68K Utilities	(disk #3)	1
52.01	Y	Math & Electronics	(disk #2)	1
53.00	Y	68K Utilities	(disk #4)	1
54.00	Y	File Maintenance	(disk #4)	1
55.00	Y	Text Processing Utils (QED for L2)	(disk #4)	4
56.00	Y	Data Base Management (SDB for L2)		4



1) All of the above volumes are available in Microware standard, Atari ST 3.5", TRS-80 Color Computer and OS-9 UG/Mizar formats. When ordering, be sure to specify only formats that you KNOW you can read on your computer! Please remember that some volumes of the Library will not fit on all formats of disk. If you do not specify the format you desire, you will be shipped either OS-9 UG standard 5.25" format (i.e., "TYPE "D"), or the format we have on file or you (as specified by you on your membership/renewal application). Please note that 8" disks are no longer available directly from the UG.

2) Volumes which are not mentioned above, or are marked with a "N" in the "Done?" column, are NOT AVAILABLE at his time. Orders for unavailable volumes will not be processed.

3) Send orders to the main UG address, making sure the envelope is clearly marked "ATTN: DISK ORDERS". Orders marked anything other than "DISK ORDERS" may be delayed an additional 4 to 6 weeks. DO NOT ADDRESS YOUR ORDER (OR CORRESPONDENCE ABOUT YOUR ORDER) TO "LIBRARY" OR "LIBRARIAN".

4) The ENTIRE OS-9 UG Library is presently available in a special multiple set of 80 track (96 tpi) double-sided double-density OS-9 format disks (UG format code "#6" ONLY). Please note that this set contains all software that is presently contained in the UG Library EXCEPT he spelling dictionary, which is ONLY available on the ndividual Volume #2 library disk.

Orders will be accepted from MEMBERS ONLY.

OS-9 Signals

Signals are one of the four means of interprocess communications available in the OS-9 system. Signals are a single, sixteen-bit quantity primarily used as a synchronizing or controlling mechanism between two processes. They are not intended as a means of exchanging data. Most frequently they are used in controlling the execution of multiple, asynchronous processes, or in coordinating the use of some shared resource. For example, this could be a device such as a printer, or a common data area in memory.

All processes that receive signals must provide a means for 'catching' each signal. This method for catching signals is termed an intercept routine. If a signal is sent to a process not providing an intercept routine, the process will be aborted. The procedures for establishing an intercept routine will be examined later.

Signal Values and Reserved Signals

OS-9 makes extensive use of signals for its own operation and provides several predefined signals. These are signals with the values 0, 1, 2 and 3. These signals have an unique status relative to all other signal values and produce effects somewhat different than other signals. These signals have the following meanings:

Signal 0 is defined as the unconditional kill signal. It is not interceptable. Any process receiving it will be terminated, regardless of whether or not an intercept routine is provided. This signal is used predominately from the console to terminate runaway processes, processes that are stuck in an infinite loop, impossible I/O tasks or other non-recoverable situations. Use of the Shell command KILL will result in this signal being sent to selected processes. Its use is not limited to the keyboard, as it can also be sent from within a user program.

Signal 1 is defined as the wake-up signal. Signals of this type are sent to processes which are sleeping (e.g. as a result of executing the FSSleep system call). This is frequently done by drivers waiting for an I/O operation to be completed. In this case, an interrupt is generated when the transfer occurs. Then the IRQ service routine will send a wake-up to the driver requesting the operation. Another example is provided by the Events system. Events use the wake-up signal to indicate to waiting (sleeping) processes that an event has changed value and may have come into range. A process does not have to have an intercept routine to be sent a wake-up signal and function properly. The wake-up signal will never kill a process, regardless of whether it provides an intercept routine or not. Sending a wake-up signal to an active process has no effect.

Signals 2 and 3 are special keyboard signals generated by entering specified keystroke sequences. Signal 2 is the keyboard abort, and is usually generated by a Control-E sequence. Signal 3 is the keyboard interrupt, and is sent in response to a Control-C sequence. These signals can be used to produce similar effects, however they are actually quite different.

Signal 2 causes the receiving process to terminate. Signal 3 may or may not cause the same process to terminate. If a process has performed I/O to the terminal from which signal 2 is sent, and the process is the last process to do so, the process will terminate. Otherwise, signal 2 causes the process to begin to operate concurrently, with respect to its Shell. This can sometimes be used to force processes into a background mode of operation.

Signals 2 and 3 are sent to the last process which has performed I/O on the terminal from which the signal is sent. Keep this is mind in order to avoid inadvertently signalling to the wrong process. Both signal 2 and 3 are interceptable. Because OS-9 utilities do not have intercept routines, the effects will be as described. User written programs with intercept routines may use these signals to their advantage. Although not presently defined, signal values of 4 to 255 are reserved for future system use. This leaves values in the range 256 - 65535 available for user purposes.

A given signal typically has only one destination. With a single exception, signals are sent between two processes. With the same exception, a signal may not be sent to more than one destination at a time. A process may however signal to multiple destinations, and a single process may receive signals from more than one process. The exception to the above description is a signal sent to a process with a Process ID (PID), of zero. This will broadcast the signal to all waiting processes which have the same group/user ID number.

Using signals

Sending a signal is relatively simple. The process ID of the receiving process must be either known or determined. If the signalling process has used the FSFork system call to create the receiving process, the signalling process can store its own ID for later use. A process may determine it's own ID by using the FSID call and exchange it with other processes via a mechanism such as a named pipe.

Once the ID of the receiver is known, **the F$Send system call is used to actuality send the signal.** This call is packaged into the C kill() function. Whether called directly or as a C function, the appropriate error checking should be performed on return from the call. Since signals do not queue on the receiver's end, attempts to send a signal to a process with a pending, unprocessed signal will return the ESUSigP error. In this event, the signalling process should wait for a short period of time (i.e. sleep for a few ticks), then retry the signalling. Attempts to signal to an invalid process ID will result in the ESIPrcID error. Care should be taken to always test for these errors when attempting to signal to another process.

The receiving process uses the F$icpt call, *via the intercept() function in C,* to catch the signal. The address of the signal handler routine to be performed when a signal occurs is part of the call. This routine must be as short and fast as possible. Typically, the only action taken is to set a global variable to the value of the received signal, then returning. In particular, calling other functions from within the intercept routine is very bad practice. If another signal is received while in the intercept function, it will be called recursively, using seventy bytes of user stack space to save the current configuration.

Once the signal handler has intercepted the signal and returned its value, the actual processing of the signal occurs. For example, the C "switch" structure might he used to select among several courses of action, based on the value of the signal received. The actual processing performed would be dependent upon the purpose of the program, hut in any case would not be performed inside the signal handler itself.

Editors notes:

The above article is a brief white paper that I received with some other Microware literature. I have more of these if you'd like to see them.

I'd like to add two items to this discussion. First, signals can be used with Basic09. In fact, they are easier to use with Basic09 than with C. You simply use the ***ON ERROR GOTO*** *statement to set the trap, and check for the error number within your error handler. After each signal another ON ERROR GOTO must be executed.. that's all there is to it.*

As for error 2 and 3 *being fatal to your Basic09 program, you have two options: stop them from occuring... or catch them. Since they come from the keyboard, you can stop them from occuring by redefining them to some character that the keyboard cannot produce. You have two ways of doing this also. One: just tmode the path:*

```
SHELL "tmode eof=255"
```

Or you can read in the path descriptor, (using syscall), and modify it to suit your needs, then read it back to the system. The advantage of this method is that by saving a copy of the 'old' PD, you can restore it when your program exits.

One more thing, notice the system overhead talked about in the paper. Note also that if two signals occur in the time frame needed to process one, the second is lost. This is because signals are intended to be used to signify EVENTS, which are relatively rare occurances. Signals should NEVER be used as substitutes for interrupts, and ***you should not get into a situation where there is a one-to-one correspondence between interrupts and signals.*** *Take note of that you terminal program writers out there!*

Coming May 15th:

RAVE
Real-Time Audio/Video Environment

Advertise in the OS-9 Users Group Newsletter! The newsletter will be printed periodically in either an 8.5" x 11" (letter size) format, or a 11" x 14.5" (tabloid size) format. The ad cost is the same regardless of publishing format, with the exception that two color ads will only be available in the issues published in the larger format. Contact a UG officer before publication deadline for information about which format the next issue will be in.

Send your camera-ready, or electronic ad copy and a check for payment to the OS-9 Users Group so that it is received no later than the 15th of the month prior to publication month.

Advertising rates are as follows (as of March , 1989):

Full Page: $200, Half Page: $100, Quarter Page: $75, Eight Page: $30

See "submisssions" for acceptable formats.

Each member is entitled to place reasonable classified ads free.

MOTD is published monthly, or at other intervals as required, by the OS9 Users Group Suite R-237, 1715 East Fowler Ave. Tampa FL 33612.

Editor: William L. Brady
1503-I Flanders Lane
Harwood, MD 20776
301-952-1761.

Make checks payable to:
"The OS9 Users Group".

Distribution is Free to members of the OS9 Users Group. Non-Members may subscribe by sending a letter ATTN Membership, at the adress above, and by paying a $25.00 annual fee.

The OS9 Users Group is not affiliated with any other organization.

This issue of the MOTD was produced using Ready Set Go! 4.5 on 2 meg Macintosh computers, a Mac II and a Plus. Mastering done on the Plus with a GCC Personal Laser Printer. Drawings produced in CANVAS 2.0. Scans via AppleScan, placed in PICT form. Fonts are Bookman, Cooper Black, Courier, and Helvitica. (6-32 points). Chief asst helpers: Jane Larwee aka: Mrs. Brady, and Larry Myers.

The date on your mail label indicates when your current dues expire. Please submit your renewal in a timely fashion.

The International Newsletter of the OS-9 Users Group
March/April 1989

BULK RATE
U.S. POSTAGE PAID
HARWOOD, MD.
PERMIT **#9**

MOTD
MESSAGE OF THE DAY

Address Correction Requested (to the Tampa Address Please)